TRIAI ,S ANI) TRIBUI ,ATION OF IMPI ,EMENTING 1 NTRANET TECHNOLOGY: 'J']] E CA SSINI INFORMATION ACCESS SYSTEM

Marc 1) Abrahams
Gerardo Rivera
Charles K Ames

Jet Propulsion 1 ,aboratory
4800 Oak Grove Drive
Pasadena, CA 91109-8099

ABSTRACT

The Cassini Information Access System (I AS) was built to provide Cassini integration and Test engineers with ready access to current technical and logistical project information. World Wide Web (WWW) technology was used to provide a consistent user interface to pre-existing interactive systems as well as new repositories and interactive services developed specificall y for Cassini. A Web "gateway" was developed for cacb pre-exi sting data service. I ¦ach service had its unique requirements including read only and read and write access to SQL databases. Other requirements called for development of many background processes to handle the access of data produced by proprietary databases and non-M abase data reposit ories. Through the use of the. W WW, anonymous ftp, Novell services, and unique data conversion programs, we Were able to put **the** system into limited operation by August 1995 and full operation by March 1996.

TECHNICAL APPROACH

With the emergence of the WWW as an information store house and with its wide use and increasing availability around the world, the IAS found itself a clear candidate for its use. The technical approach taken with lAS was to use the World Wide Web to provide a uniform interface to a numl)er of pre-exi st ing information services. This approach allowed access to a variet y of informat ion sources and accessibility from multiple platforms

BACKGROUND

The Assembl y Test and 1 ,aunch Operat ions (ATI .0) group was continually requesting up-to-date technical documents and schedules for the assembly and testing of the Cassini Spacecraft. The need for the latest release of an engineering document is vital to the successful accomplishment of testing and spacecraft assembly. in addition, published information needs to be disseminated to the other agencies, industrial partners and academia participants involved in the spacecraft assembly and test. ATI.O engineers needed intc.grated access to widely distributed documents and widely varying information systems. The information Access System was designed to gather needed technical documents and schedules using a cor nmon user interface to view the in format ion. The task began in early June 1995.

Gathering of Technical Information

Gathering the technical information became a task in itself since the technical information was developed by different groups on different plat forms and with different applications. The ATI.O test engineers constantly request new releases of documents which can delay his/her testing activity while waiting for the needed document. The current implementation created volumes of documents which if needed are stored and control led within ones own area. 1 n addit ion, data had to be submitted for review by other members of the technical team and managers. 1)isseminating the information to other agencies also became a time consuming task (*see figure 1*).

Problems With Former Process

The informat ion gathering process (lid not use the er nerging technology, and preparing every document for prompt review became a needed request from the users. The documents had to be reviewed and submitted for pub] icat ion to the other gr oups in a time.ly manner to prevent testing delays. Other agencies requesting technical

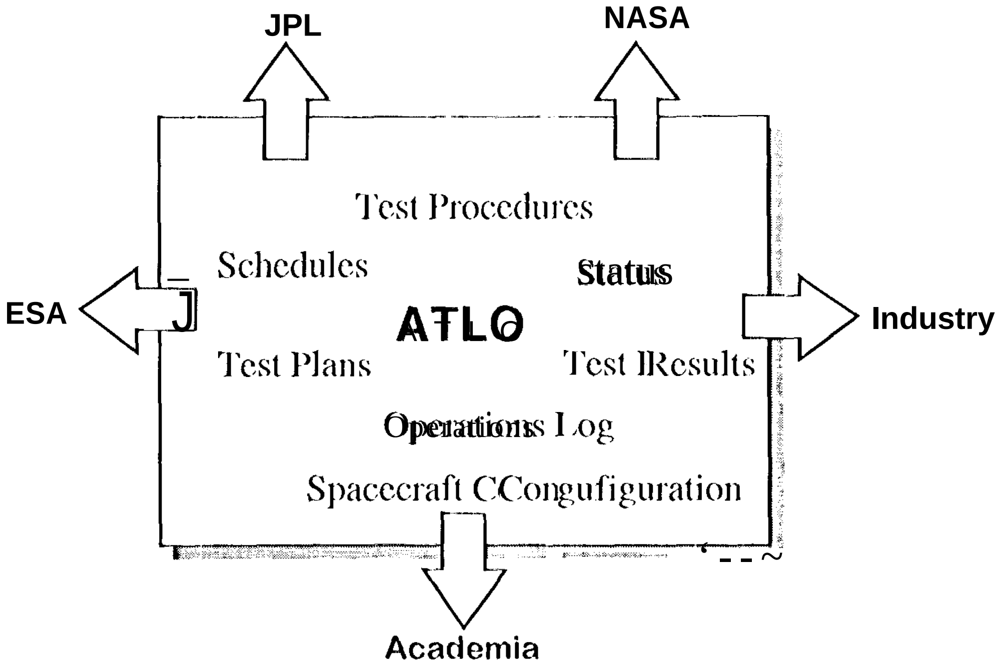


Figure 1 Outputs from ATLO

information from different parts of the country would encounter additional delays.

Possible Solutions

Possible solutions to the current implementation are:

A. Continue with the current practice and provided additional training, to the data repositories for prompt (1cI ivery of documents.

B. Maintain different application packages on ones own computer for the different formats of the technical data,

C. Use the WWW to retrieve documents and submit testing activity information.

Trade Studies On The Possible Solutions

Providing additional training would help prevent some or a few problems but would not solve the larger issue of delivering newly released documents on demand. While, maintaining different application packages on everyone's machine would be costly for the project and learning a new application for every technical document would be impracticable. Therefore, the WWW server architecture proved to be the most promising solution. The use of a common interface (web browser) and the ability to create Common Gateway Interfaces (CGI) with different sources of data and databases became a viable solution.

DESIGN

The WWW and its set of protocols would allow the exchange of dat a among large organizations and universities. MoJc.over, it would provide a means for the data to be accessed without needing to know the technical issues behind the retrieval or submission process. The user would use a web browser to interface with the data. A simple click of t 1 ic mouse but ton would retrieve the. required data and display it on the users monitor.

Data Repositories To Be Accessed

The data repositories consisted of six different databases and a directory containing]'i amc.Maker@ documents. The technical issues involved in retrieving the data were: interfacing with the data, location of the data, transferring the data, filtering the data, and access rights to the data. In many cases the data would be converted into a format that could be read by a web browser

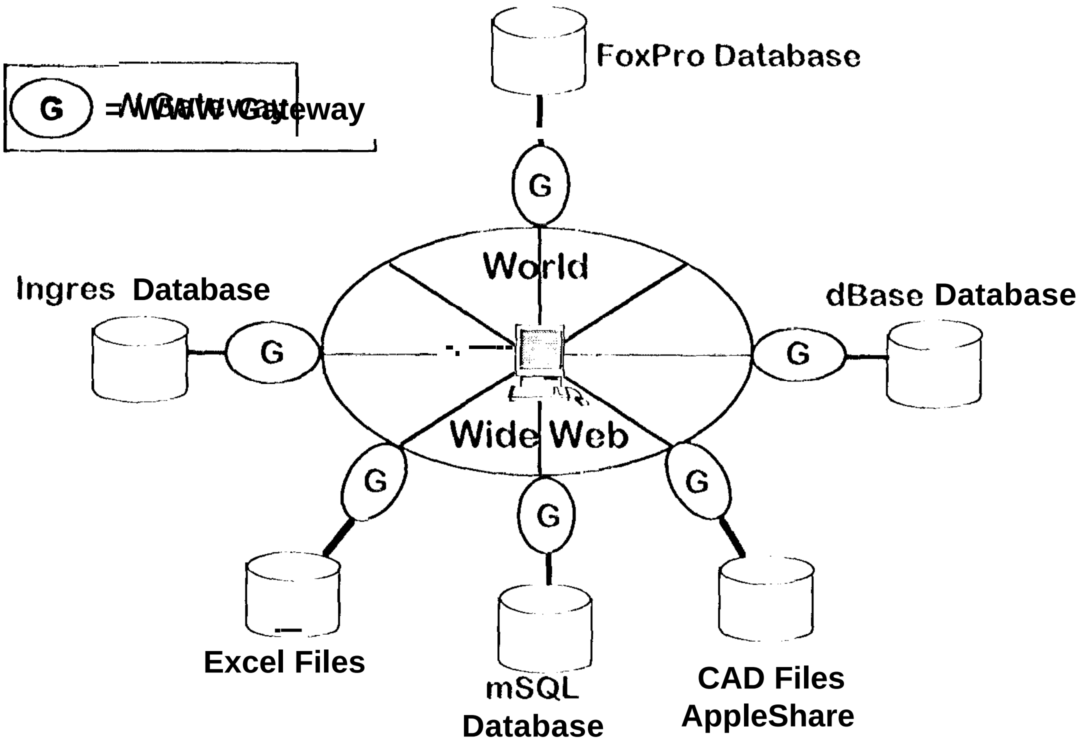


Figure 2 Gateway Interface.

or di splayed using t he Adobe'M Acrobat Reader or 17ramcMakcr 1 'rameReader®. The owners of the data would still maintain and update the data and no changes to the data would occur without correct access rights.

Interfacing With Data - A Common Gateway Interface or CGI as its commonly known was developed for each type of database as depicted in *figure 2*. The CGI is basically a program written in C, 1'10<1,, Basic or any other programming language that will support data extraction, and capable of reading/writ ing data in ASCI 1 format. The program sends commands to the database in order to extract the data. Once the data has been extracted froi n the database, the data is converted into text or 11 ypertext Markup 1 ,anguage [11'1'M I,) format. IITMI, is a tagging format for marking up documents with informational tags that dictate how text wi I I appear in a web browser. Time critical data is converted immediate.] y when the user submits a request from his/her web brow ser. Within a few seconds of the request, the data is converted and presented to the user in a readable. and understandable format. Other documents which are not time critical are also downloaded to the user. 1 lowever, these documents are not Updated to the. current release. These other documents st il 1 need to be available for review but ai c converted overnight with the, latest release avai table the next morning. These overnight documents go t hrough a special fi ltering process which wi I l be d i scussed later under filtering of data. Sti ll other data is converted by the web site administrator when a special document is requested. 1 ,arge requirement documents or lengthy test procedures are converted into Portable 1)ocument I 'ormat (f'1)F) and are viewed with A(101)c's Acrobat Reader. Other files which were created using lkamc.Maker are viewed using 1 'i ameReader. The web browser can be configured to launch (run) the Acrobat Reader or 1 'rameReader when a PI)F or Frame file is selected.

Location Of Data - The location of the data can be on the same machine your web server is located, or the data can exist cm different machines at different locations throughout your organization. The organizat ion that owns the data has to allow you access to the data either with read only or read and write privileges. Most of the data that is requested is read only and most organi zat ions will agree to allow your web services access al 1 day long. Other data which is constantly updated such as log files must have read and write privileges. These privileges will be discussed later under access rights.

Transfer Of Data - The data that is transferred depends a great deal on your network and the speed at which your network can deliver the data back to your web server. 1 'or time critical data your web server (via CGI) must go to the machine where the data resides and request the data. The machine on the other end must deliver the data back to your web server in order to present the needed data back to the requested user. Data that is not time critical can be move.d overnight when the network is least busy, 1 'or example a si7.able amount of postscript files can be generated every night and converted into PDF. It would not be practicable to transfer the files during the. (lay and tic. up the network while the files are converted. A cron (dock daemon under UNIX®) job can be set up to run overnight and transfer the needed data.

Filtering Of Data - Data goes through a filtering process for: converting live data into 1 ITML format, converting postscript files into PDF files, converting '1 'agged Image File Format ('1'1 FF) files into Graphics Interchange Format (GIF) files, or creating a Wide Area Information Service (WAIS) search index on large amounts of data. Database documents are filtered through a CGI program which interprets (parses) the special database characters and convert the characters into an 1 ITML file. The 1 ITM 1, file can then be read by a web browser. Postscript files are converted into P])F files using Adobe's Acrobat Distiller®. Converting graphics files required a graphics conversion tool. Most graphic conversion tools can be found on the WWW with the most widely usc(1 being NetPBM. '1'1 I'1 'files are converted into Portable Pix Map (PPM) files and then the PPM files are converted into G] F files. Searching on a large database can often take more time then a person is willing to wait. A WAIS index improves the sped of a search by creating a list of words that can be searched quickly. The WAIS index makes it easier to find the needed information without slowing down the web server.

Access Rights - Access rights to the data has to be closely watched since vital information of a project cannot be released until final analysis of the data is completed. Your web site has to acquire the. comet access rights from the owner's of the data, and the accounts and directory must be established. 1'01 read only data this is very simple but for data that is constantly changing a much tighter restriction must be placed. A username and password file must be set up in order to protect vital information. When a user rc.quests vital information from his or her web browser, the web server can be configured to authenticate against the user requesting the information. Access control on a web server can be set up by hostname, Internet Protocol (11') address, user or group. In addition, various encryption schemes are available for the web servers. A person's username and password can be encrypted before the information is sent along the network or across the country.

IMPLEMENTATION

For the Cassini 1 AS project two web servers were employed to handle CGI programs, user interfaces and user authentication. One of the servers ran on a UNIX machine and the other server ran on a Windows® machine. in addition, other network services (Novc11,] 'TP) were used for data transfer and storage. Some of the other services wc initially requested access to decided to create the.ir own web services and provide their own authentication. This made it easier since only a link to their service was required. Schedules for the Cassini project were developed on Macintosh® applications and updated periodically, These schedules were saved as postscript files by the author. A cron job would convert the postscript file.s into PDF files and the files would be stored on the UNIX machine. Circuit data sheet files created in Canvas™ were also converted into PDI files.] lowever, the files had to be converted with a s])ecial program because the circuit data sheets were to large to view as a regular document. '1 'he program first opened the. Canvas file and reduced it to fit on a 8.5 by 11 inch paper. The program the.n saved the. Canvas file as a postscript file and from there the postscript file was converted into PDF. 1 .ive data (constantly changing) was stored on a (1 I last database. and converted to HTML format when the user requested a document. A CGI program (written in visual basic) would parse the database and determine which special characters were used (tabs, 1 inefeeds, new] ines and tables). The CGI program al so determined links to other documents within the database and created links to those documents. The final result of the CGI program was an HTML file. that contained the correct information and reference l inks.]J addition, a cron job would convert TIFF files into GIF files and the GIF files would be inserted into the HTML document. All the HTML conversion occurred in a manner of seconds and ran on a Windows machine. The graphic conversion ran on a UNIX machine overnight and used a mirror ftp program to copy only the latest released graphic files. Another service that usc(1 Jive data, saved their file as a Microsoft Excel document. '1'he 1 ;xcel document would be copied from the Jive data and an Excel macro would be usc(1 to extract the needed data. The data would then be displayed to the user as an HTML table listing. 'Jim final living document used a form driven web di spl ay. The form displayed a daily report of test activities for that day or previous days. The form would cal l a CGI program that would save the information in a mSQL database. The database could then be queried and the informat ion would be di splayed as an HTM I, table. The twist to the. ent ry forn n was a username and password authentication system that prevented just anyone from entering informat ion.

PROBLEMS

Several problems were encountered while developing the Cassini 1 AS. None became major problems and most of the problems were solved by trying a different approach or setting configuration files. The problems encounter included; special characters in the database continued to change, disk space for converting documents started to decrease, the net work would go down, serve.r configurat ion files had to be set correctly and graphic conversion tools did not work on all platforms. The special program that created the 1 ITM 1. files from a dBase database did not work properly because special characters within the database would change. '1 'he program would interpret special characters and would define the. structure of the HTML file., however, the person that started inputting the data in the database decided to use a different character for creating tables. This would cause new tables to be displayed incorrectly on the web browser. The problem was fixed by adding addit ional lines of code to the! program to check for the new character. Another area of concern was disk space. The required disk space was underest imated since files were larger t han ant icipated. The problem was fixed by partit ioning more space for the. 1 AS project. The next item of concern was the network. The network would go down and services connected to the servers would be disconnected. Most of the time resetting the server would fix the problem but sometimes one would have to wait for the network to come backup before met ting it, The network problem was very rare and most of the time service was available. The server configuration files had to be changed in order to support PDF files. A PDF file would be downloaded to the persons compute.J" and a helper application like AcrobatReader would open the file.. 1 lowever, if the mine type is not set correctly in the server's configurat ion file, the web server would attempt to send the file to the web browser and the web browser would not recognize the file. The circuit data sheets that were created in Canvas had to be converted into PDF files. Originally, the Macintosh was going to be used to convert the Canvas files however, scripting on the Macintosh could not be accomplished and the problem was corrected by using a PC to execute the req uired commands.

CONCLUSION

'Jim Cassini 1 AS project looked at the emerging technologies and applied thcm to a large organizational net work. It centralized the needed services requested by its users and it provided a quick tam around. Most of the owners of the data continued to maintain the data, and for the most part they continued to operate in the. same manner as before. This Cassi ni I AS project started in cad y June 1995 and was up and running by August 1 995. It went into full operation by March 1996 and is currently maintained for update.s. 'The Information Access System has been very well received at JPL. Broader applications of IAS for other projects and for administration services are being considered.

Trials and Tribulation of Implementing Intranet Technology:

Cassini Information Access System

10/29/96

Information Access System (IAS)

Marc D Abrahams
Charles K Ames
Gerardo Rivera

Information Access System

- ❖ Technical Approach
- ❖ Background
- ❖ Design
- ❖ Implementation
- ❖ Problems
- ❖ Conclusion

Technical Approach

- Use the World Wide Web (WWW)
  - Pre-existing information services
- Accessibility From Multiple Platforms
  - UNIX
  - Windows 3.1, NT, 95
  - Macintosh

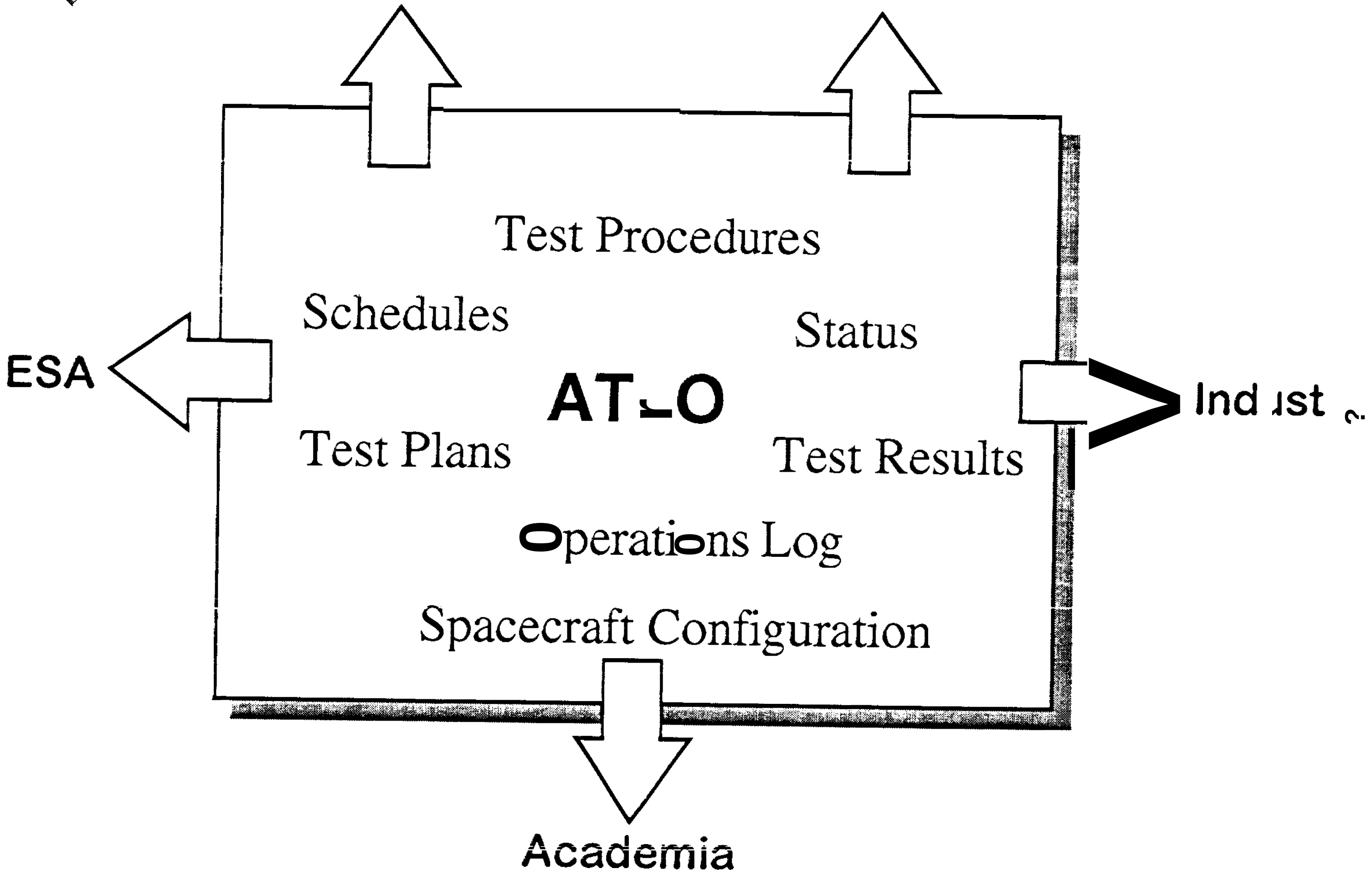
Background

- Cassini Spacecraft
- Gathering of Technical Information
- Problems with Former Process
- Possible Solutions
- Trade Studies on the Possible Solutions

Background

- Cassini Spacecraft
  - Testing Group Request for up-to-date data
  - International Partners, Industrial Partners and Academia
- Gathering of Technical Information
  - Different Groups and Platforms
  - Reviews by team members and managers
  - Disseminating information

Background



JPL

NASA

Test Procedures

Schedules

Status

ESA

ATLO

Ind ıst ?

Test Plans

Test Results

Operations Log

Spacecraft Configuration

Academia

10/29/96

Background

- Problems with Former Process
  - Almost no use of emerging technology
  - Reviewed in a timely manner
  - Requests for technical information
- Possible Solutions
  - Continue with current practice
  - Maintain different application packages
  - Use WWW to retrieve data

Background

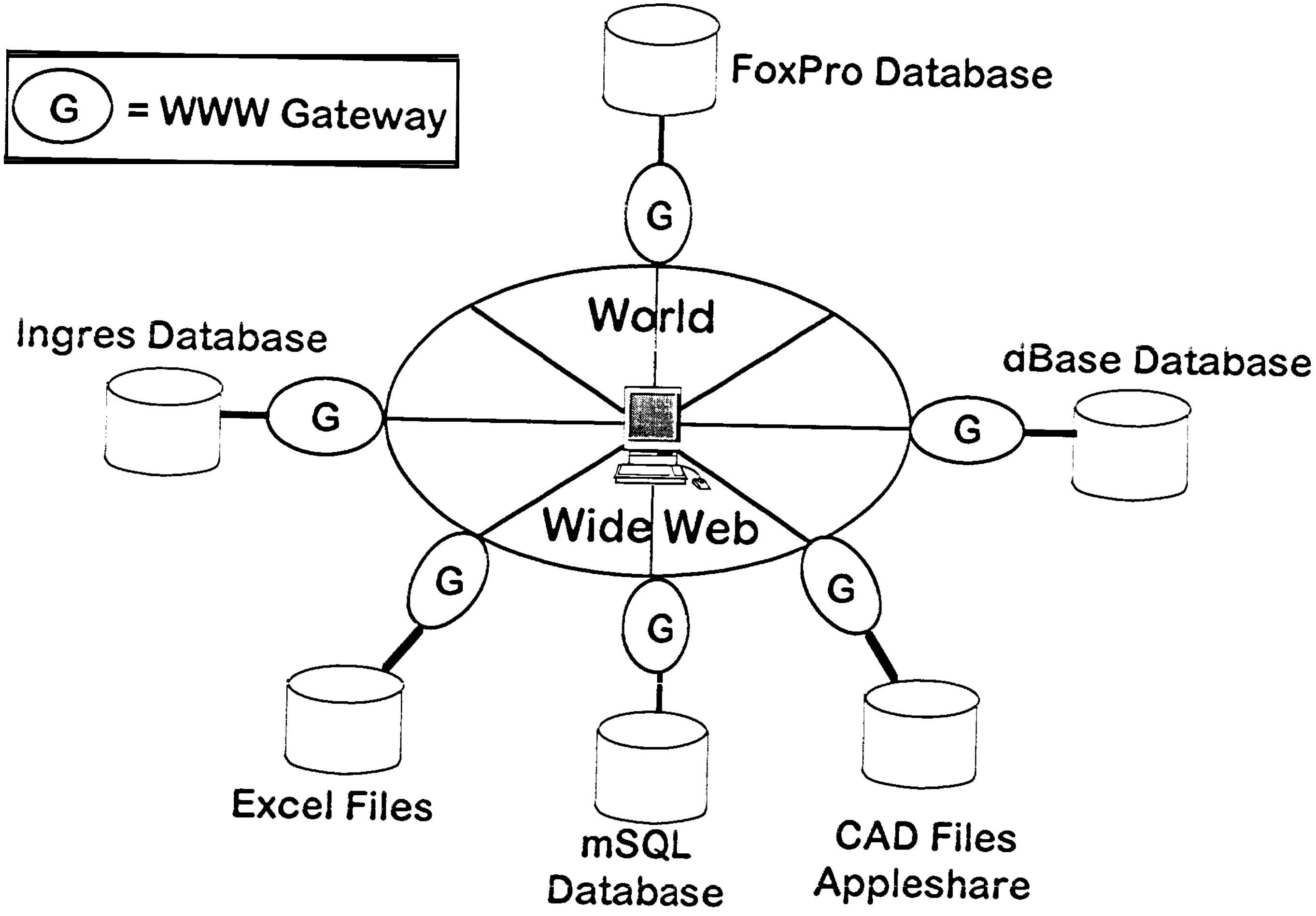
- Trade Studies on the Possible Solutions
  - Provide additional training
  - Maintaining different applications
  - Common interface

Design

- Use WWW
- No programming skills needed
- Use a web browser
- Access six different databases

10/29/96

DCDsign



G = WWW Gateway

World Wide Web

FoxPro Database

Ingres Database

dBase Database

Excel Files

mSQL Database

CAD Files Appleshare

10/29/96

Design

- ❖ Technical Issues in Retrieving Data
  - – Interface with data
  - – Location of data
  - – Transfer of data
  - – Filtering data
  - – Access rights to data

Design

- ❖ Interface with data
  - – Build Common Gateway Interfaces (CGIs)
    - ◆ C, PERL, Basic....
  - – Convert Data into Hypertext Markup Language (HTML)
  - – Overnight Conversion
  - – Convert Large documents
- ❖ Location of data

Design

- Transfer of data
  - Network speed
  - Now or overnight
- Filtering data
  - Convert to HTML, Portable Document Format (PDF) or graphics
  - Wide Area Information Service (WAIS) search

Design

- ❖ Access rights to data
  - – Close watch on vital information
  - – Establish accounts
  - – Encrypt data

Implementation

- Web Servers
  - UNIX
  - Windows
- Access to other Services
- Overnight Conversion
- CGI Programming
- Graphics Conversion